Silicon Valley's Top Investors Bet on This 22-Year-Old Founder | Sola, Jessica Wu

I was actually the youngest quant researcher at a pretty big hedge fund today and so had started my career in finance but had seen a lot of different areas very thesis driven to just pure numbers and numbers crunching. You know when I was working in corporate finance I would work uh I don't know 9 10 hours a day and uh 5 days a week and then if I was called in on a weekend I would honestly be very miserable. It just felt like it it wasn't my own thing. Now I work well every waking second uh and 7 days a week, but I've never been happier. And I think I think I'm young, so that's one piece that gives me a little bit more optionality. When it comes to what you want to do, though, this sounds very simple, but it's really just what feels right. The team feels that way, too. It's really like when you own a piece of something and you're you're really growing it from ground up, it's easy to feel excited. And so my advice for like stable career versus startup is if you have the optionality to I think you know whatever makes you feel happy, whatever makes you feel like you're excited to go to work every single day and that you feel very fulfilled in what you're doing. Hi, I'm Jessica, co-founder and CEO of Sol. Sol is an agentic process automation platform. We help businesses automate their most critical and operational workflows using AI in a much easier and faster way than traditional RPA. Solus started out of YC about 2 years ago. Since then, we have raised a seed round led by Sarah at Conviction. Most recently, we raised a series A led by A16Z following an enormous amount of traction from our enterprise customers. We have 5xed our revenue this year. Execution volume on the platform is doubling month over month since the beginning of this year. And we're proud to partner with some of the largest companies in the world, including the Fortune 100, Amlaw 100, and some of the largest private businesses in logistics and healthcare. Uh, I grew up doing a lot of very competitive things. Played competitive piano and I did competitive math and things like that. I think that gives you a pretty hard will. It probably makes you very disciplined. I think I'm good at taking risks and putting myself out in places that I don't feel comfortable. And I think that builds up a lot of resilience. Like doing something even if you don't feel like you're 100% ready. I do that a lot. And just jumping into things and really going for it. I think that builds up a lot of character. When I was in high school, I had visited MIT a couple of times. I think it does a great job of the whole, you know, you're the dumbest person in the room kind of thing. It's always the most fun when you're surrounded by a lot of people you can learn from. I think MIT is a very unique place. It's the only place where it's really exactly as it portrayed in the movies. You have

people building roller coasters in the front lawn and you have people training models in the dorm basement and things like that and it's every bit as real as as it's described. I think that MIT really puts a ceiling on how technically difficult things get. I've been working on this company for about two years now and have experienced a lot of different things since college, but never have I had to use like so many brain cells to just think about a really hard problem. I think the most important thing you should do is jump into everything as quickly as you can. And MIT does a great job of showing you exactly what's on the forefront of what you should be paying attention to. You should absolutely go learn about all the new things that people are doing research about. You should absolutely spend as much time as you can with people that are doing said research. And I think that if you create a habit around trying to learn as much as you can and intake as much what's on the frontier as possible, then you're better equipped, especially now to to learn and to work in an environment like today where things change so rapidly and there's so many things happening in the tech world. I think the thing that you get from just a very technical background is you can break things down very easily. If you can communicate what you're trying to solve in a very simple way, it makes it a lot easier to solve and that just gives you a really good framework around solving things, especially in a landscape like today where new model basically comes out every single week and things are constantly changing. I think there's a lot of value in always centering back to like what are you trying to solve for your customer or for your user, whoever that is, and then just building on top of that. And I think once you look and sort of narrow things down to the root problem, not just the problem that people are saying, then you can deliver an enormous amount of value to the people that are interfacing with what you're doing. Before I worked on Solola as a startup, I had no idea that I wanted to do a startup. I had tried a lot of different things. I had worked in venture. I had worked at a couple hedge funds. I was actually the youngest quant researcher at a pretty big hedge fund today. Finance was an interesting choice. I had an adviser who recommended to me that it's actually pretty good to spend some time in finance because it teaches you a way of thinking that's pretty special, especially if you're doing trading. Teaches you to be very objective, that you should always calculate odds for things. I think I am someone who's guilty of not thinking in a very numerical or like standardized way all the time. And so I think working in finance for a couple years gave me a lot of perspective on being objective, being able to calculate and just think through things in a very rational way. And that's something that stuck with me. Doing a startup can be very emotional and I think always going back to first

principles thinking and trying to think in a more like statistical way can be pretty helpful in terms of making decisions even like in a startup lens. Maybe you're looking at a super huge deployment and that's something you have to really weigh or someone is really looking at features like all of those things can be mapped out in a way that's more rational. You can say, you know, what are the odds that this particular, you know, this isn't perfect math. This particular customer converts by this particular day. What are the things that I'm trading off? And it gives you a very statistical framework to think about these decisions without just I really like this customer, so I want to work with them. People tell you that you can learn a lot from playing poker and that applies a lot to the startup world. I think that's true. A lot of it is just odds, being rational, and then, you know, making decisions in the best direction. And it's actually very useful when you work on a startup. At one of the hedge funds we worked at, they had a lot of manual work around interacting with this really old brokerage software that they built on top of. At the time, I had been using RPA tools to try to automate that. I didn't really know how big the space of RPA was. So, RPA stands for robotic process automation. RPA tools basically automate manual work by replicating workflows the way humans do. So moving the mouse, typing things in, clicking, and interfacing with browser and desktop applications the way humans do. What I did know was that there is a lot of manual work at these larger companies. There's no good tools out there that are really easy to use. And even though I have a computer science background, it was still really hard for me to just build a simple browser or desktop automation. That was sort of where the the seat of the problem happened. My co-founder actually had a really similar experience. He was building out hospital systems at MGH. they had used some very old tooling and he was also surprised by how brittle it was and how hard it was to implement. I think those were really good glimpses for us into what real world work looks like. Um, when you're at MIT, you live in a very tech-forward world and you're surrounded by tools that are very easy to use, that everything has APIs that plug into each other, you can build automations very easily. But in the real world, when you're looking at most companies out there, they're doing really manual work. People operate across a ton of different systems that don't connect. As you can imagine, people doing operations will touch spreadsheets, internal portals, external systems, they will touch files, and just about everything in between. Those were two really lucky glimpses I think that we had into what that actually looked like. From there, when we went through YC, like I mentioned, we kind of came in without too much of an idea. We spent about the first month or so just figuring out what we wanted to

build. I think the RPA space was very exciting to us because we understood very well what that problem looked like and how we wanted to solve it. Obviously RPA is very big and I think that piece is exciting to us. The ambition of automating all digital work is a perennial and obvious one and I think the aspects of like being able to marry AI in the real world for enterprise companies and all the technical and model improvements on the flip side of things was a very exciting sort of combination for us. Solo's MVP was very simple. We had a recorder just like today. That part's the same that lets you take a recording of your workflow. You could upload it, but uh different from today. Didn't really do a great job of laying out what your workflow did. You could also run it and it would just run on your own computer and repeat back the same steps. Not quite as intelligent or easy to use as we had hoped. And since then, we've kept a lot of things the same. So, you know, you can still record a workflow, you can still upload it to the platform, but now you can run it hundreds of VMs at scale. You can edit it and change things in very fine granularity. You can add logic. You can add information on top. And you have a lot of optionality on how you want to orchestrate and run these things at scale. One thing we had probably underestimated was just there were a lot of trade-offs that we had to make. But it it is a big question of, you know, existing tools offer everything under the sun and they've been building this tool for 20 years. Like how do you actually get up to speed? One thing that we found to be a pretty big balancing act is just getting to the point where we can be helpful for customers but also, you know, being able to do so in a pretty timely fashion. There are a couple hard moments that I can think of in the early days where we had to give up really exciting early revenue. It was just a point where we had to really focus on building this core product, but we had to turn down customers the very beginning. Turn down people when we were, you know, almost there, but not quite. And that was that was very difficult. I think in the early months you get pulled in a lot of different directions. There were some hard conversations where we had to say no to some huge names that we were very excited about. It didn't feel aligned with the direction that we wanted to go down. We were capacity constrained and it would mean giving up or sacrificing too many other things and other areas of the business. I'm pretty big on advisors. I'm someone that asks for a lot of advice. I like to surround myself with people that I think have a lot more experience. So, whenever I get panicked or I'm under pressure, the first thing I'll do is just go to my all of my resources and get people's takes on things. I think usually that puts me in a pretty good place. We chose YC because we had heard really good things about it and that was also the advice that we got early on, which is that you're going to be

surrounded by a lot of really amazing people and you're going to have 3 months to lock in. YC tells you to try and they really push you to try and sell something. This means that even if you don't have a working product, you can put up like a, you know, use lovable or something to put up a fake fake front end and then you should try and go sell it. And the good thing about that is a lot of the times when you build something, I don't know if this applies as much to solo, but in general when you build something, you don't know if you're going down the right direction. Most clear way you know you're building something that people want is if they'll pay for it, right? And if they'll keep paying for it. YC does a good job in really pushing you to sell even if you may not be there. You know, don't spend 6 months to build a product. First try and sell something. See if it works. You might burn a couple bridges doing that, which is why I'm saying your mileage may vary, but at least you have very clear picture of I know this person would pay for this. I know a lot of people would pay for this and they would depend on it a lot or I'm solving some really meaningful problem. First cell is not good when you get a little bigger like you know don't sell a fake feature. It still is a pretty good mindset to operate behind which is even once you're working on the right product there's a lot of directions you should go try and experiment build different mockups of things try and talk to customers get something out there and then go from there and iterate and polish it afterwards I think we met our first customer actually very early on it was around YC we were in the phase of doing the thing where we sold the product that didn't really exist the good thing was we were pretty honest with them and we told them this thing doesn't exist but it will exist in a couple months and they were like okay well come talk to us when it does I think it's really hard advice. It's very unintuitive. You know, you're taught your whole life to finish something and then go sell it or then go present it or whatever it is. And this is almost the entire opposite. It's like put up the most minimal fake version you probably can of something and then go sell it and then if it's good then go back and build it. I think it's not for the faint of heart for sure. And sometimes, you know, maybe you really succeeded and you actually sold the thing and now you have to deliver. And YC's advice there is probably like spend a week in the basement coding so that you can get it out. It's not easy advice to take and selling without actually having the thing is not very intuitive, but I think it it at least puts you on the right direction. It gives you a very clear yes or no for if this is something you should be doing at all. One of my favorite books is called Delivering Happiness. It's quite good. It's about like sort of how the little things matter and how customer delight is kind of a north north star, but it doesn't just come in one form factor of like I I did this thing

on time. There's a million other ways you can build that up. If your customers are happy, most other things fall in place is something that we've tend to found. You know, especially when you're building very critical software for a company, you're asking a company that's been around for decades or maybe even centuries to trust you to do their most critical operations. That is by no means easy. To this day, uh most of Sol's customers come through word of mouth. And that's because existing customers have felt so happy with the experience. You're small, you're early, so what can you do to make up for being a much newer tool? And that's usually one build a really great experience and really solve their problems. That's the obvious one. Second is to ship fast and really take in good feedback. This goes both ways. So you know you want to pick customers that will give you good feedback and you also want your customers or you also will want to take in the feedback that they give you and move very quickly on what they're asking. There's a lot of ways that are more traditional than just building good tools where you can win enterprise companies over. You can be very supportive. You can listen to people. Listening is the most important thing you can do for customers. For some of our early design partners, we still do. We hop on a weekly call with them. We basically take in their feedback week over week and then we try to deliver at a pretty fast rate. And that's the sort of thing that will make people pick the startup over the really old legacy incumbent. Aside from just the tool being better, one of our largest customers wanted to do a deployment of Sol last year. They wanted to do it during the most quiet period of the year, which was on Christmas day because they don't get a ton of volume then. I hope that we don't have to ask this of the team again. We all came in, we worked that week. They had been using Solo for a long time and they had been championing it internally and were very excited. We were like, "Okay, if the deployment is going to happen on Christmas, like we're going to do it." There have been moments like that. I think in general, we try to really go above and beyond. People are trusting us to run their operations. So, they'll use sola to send their invoices and do their accounts payable or they'll use it to actually send out shipments or they will enter in patient data. So all these things are very critical like uh one thing that we realized when we built solar is like we can never go down. That actually means that a real business out there who has hundreds or thousands of employees like their their billing is going to be late that day. We've done a lot to basically maintain that. It's one of those things where it's a little bit different from like a consumer tool or something that's AI that's generating copy or you know finding sales leads like these things. If your tool goes down it's a big deal and it disrupts people's work but it's not the end of the world. the

nature of our business when we go down is sort of the end of the world. So, we've done a lot for our customers. We stay up and running all the time. We've spent a million weekends and a million different holidays in the office. But, I think along the way, it's it's been good. Like, there's a little bit of, you know, we're extremely thankful to the people who trusted us to take care of these workflows and made the bet on an early new startup. And hopefully, we can continue to deliver on everything that we've promised. I think building a startup is really really unimaginably difficult. You're on a roller coaster and you're constantly experiencing very high highs and very low lows and then it's also compressed and so it's happening, you know, morning might be your best day ever and then 2 hours later you're suffering from probably the worst news you've ever heard. And this happens uh pretty much every day. um maybe every week, maybe not like an exact moment, but I think as a founder, you really just learn to you learn that things sort of return to the norm. You have this comfort in your head that things will be okay even if they don't feel okay because they always end up being okay provided that you are building in the right market and also that you have the right people around you. And with those two things, you might have some huge incident happens. Really bad news like someone leaves the team. Maybe a customer is your champion who you were selling to left and in the middle of the sales process. Like these things feel really bad. And then once you get a little bit more used to them and they've been happening so often, you learn to take a step back and you're kind of able to process a lot more of the things happening at a startup um than before. And I think the early team also kind of learns that as well and be able to look at the big picture and that tends to be much more helpful in terms of just mental piece when you're doing this. What we're trying to build is a really big mission. So there will be no problem of us having 10 years to spend on it. I think it's an enormous problem space. The other is that I I really do believe in what Solola is trying to accomplish. Um, I think that there's a lot of work that people shouldn't have to do that's very manual and very tedious. And if we can free people up, they can do things that are very fulfilling and creative and very interesting. And I hope that, you know, businesses can start running on sola and relying on sola to do the things that are manual and wrote and that people don't really want to do and instead spend time on things that are exciting and think through leadership and strategy and decision-m and all the more interesting parts of work.